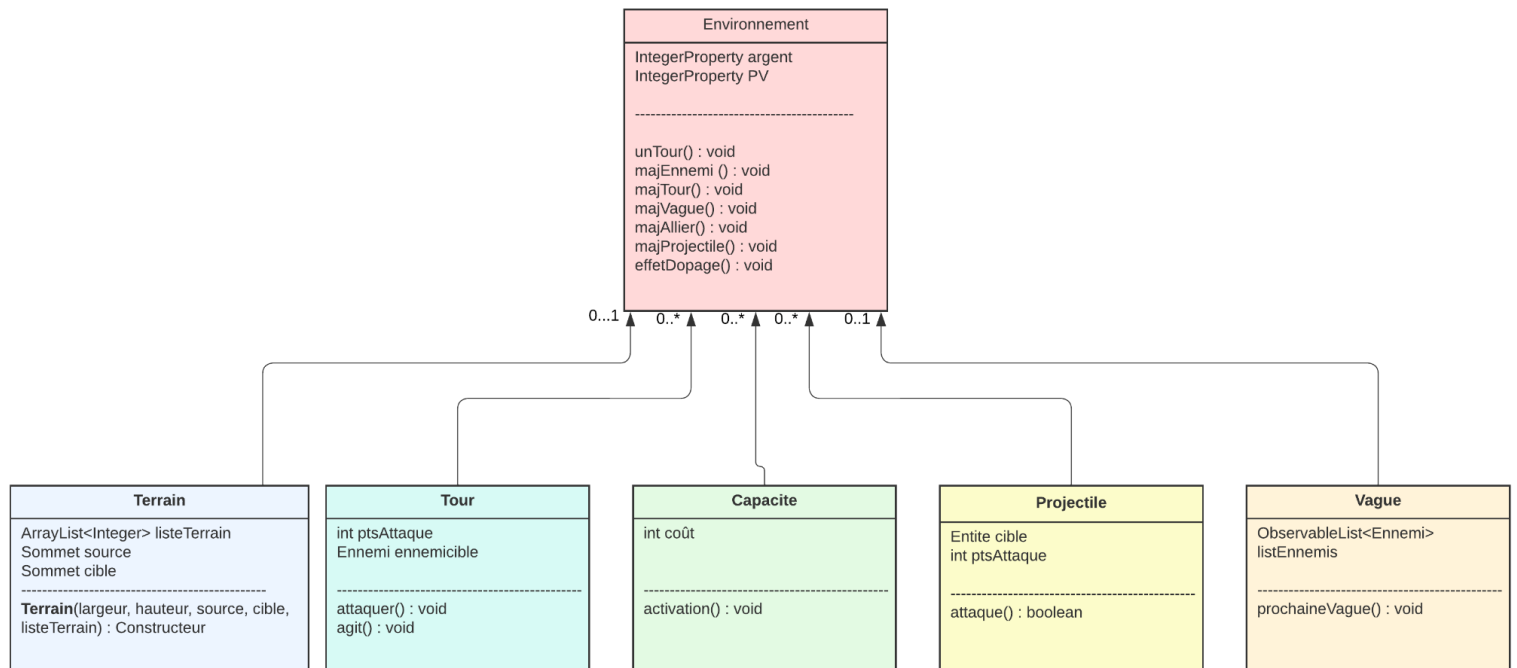


SAE I.H.M

Document Diagrammes de Classe:

- Premier diagramme de classe:

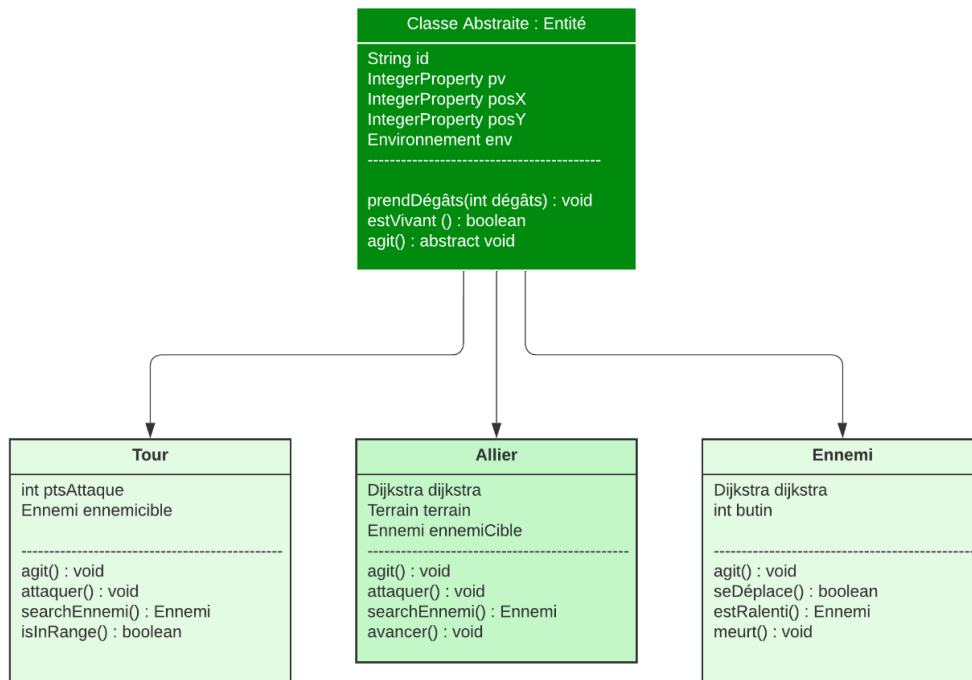


Ce diagramme est la pierre angulaire du fonctionnement de notre modèle. En effet, ce diagramme présente les liens entretenus par l'**Environnement** avec les différentes classes du modèle.

La classe **Environnement** ci-dessus, s'occupe de la gestion des différents acteurs. Elle les fait agir, les met à jour à chaque tour et s'actualise ainsi.

La classe **Environnement** tient son intérêt dans la programmation orientée objets dans le fait qu'elle incarne le moteur du jeu entre plusieurs entités. Sa responsabilité reste limitée à la gestion de groupes d'acteurs.

- Deuxième diagramme de classe :

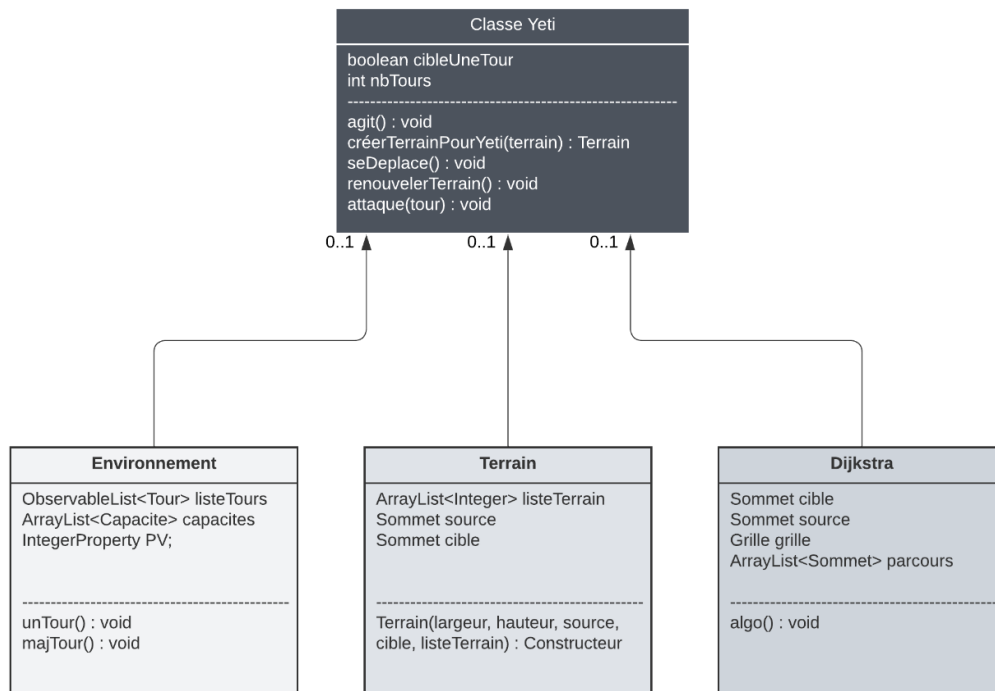


Ce diagramme présente l'héritage principal de notre modèle. En d'autres termes, la classe **Entité** incarne tous les acteurs de notre modèle. Cette classe est très utile dans la chasse à la redondance du code.

On remarque ci-dessus que les classes **Tour**, **Allier** et **Ennemi** héritent toutes de la classe abstraite **Entité**. Ce diagramme souligne aussi les différences de chaque sous-classe à travers leurs méthodes et attributs. Les **Tours** attaquent, les **Ennemis** se déplacent et les **Alliés** se déplacent et attaquent.

D'un point de vue programmation orientée objets, dans ce cas précis on comprend l'intérêt d'un héritage au préjudice du polymorphisme.

- Troisième diagramme de classe :



Le diagramme ci-dessus nous servira à vous présenter le fonctionnement d'un Ennemi type, en l'occurrence le **Yéti**. Cet ennemi type est intéressant dans sa composition, il possède comme attributs trois classes, l'**Environnement**, le **Terrain** et le **Dijkstra**.

La classe **Environnement** permet au **Yéti** d'avoir connaissance des Tours présentes en son sein et de la position de l'objectif sur la map pour s'y diriger s'il n'y a pas de tours présentes.

Le **Yéti** se déplace grâce au parcours de son attribut **Dijkstra**, calculé depuis le **Terrain**.

Le terrain du **Yéti** est différent de tous les autres ennemis, il peut se déplacer partout où il y a de la neige. Il doit donc se construire son propre terrain via une méthode (`créerTerrainPourYéti`).

A travers ses attributs de différentes classes, il délègue la charge de travail nécessaire à son fonctionnement, ce qui d'un point de vue orienté objets est intéressant.